Robot Learning

Coursework 2

CID: 02472956

Part 1: Description of your Method

I developed a robot learning algorithm using a hybrid approach that incorporates both open-loop and closed-loop model-based methods. The core of this method involves initializing a dynamic model with six layers to predict future states based on current state-action pairs. The planning function is designed to sample 100 possible actions, selects the optimal one based on proximity to the baseline action (state – goal state), and incorporates noise for realism. This process is iteratively applied to sample a 200-step best path, which is then executed with dynamic action adjustments to navigate complex terrains and avoid getting stuck. This optimal next action, with added noise, is then executed from the current state, stored in the buffer, and the model is trained once the current path is completed, or goal reached. The training involves updating the model with small batches of data from buffer, with 100 epochs per training session.

The executed training action adapts between open and closed loop strategies based on terrain complexity, optimizing path planning and execution in real-time. The robot is encouraged to utilize pre-planned actions from the open loop in darker areas (valleys), as these are assumed to be less complex terrains. However, when encountering mountains (light areas), the robot switches to the closed loop to replan the path and actions from the current state for each state.

Both testing and training follow the same principle for next action determination. However, during testing, without direct feedback on the next state, the "getting stuck" condition changes to evaluate the difference between the proposed action and the baseline action. If the difference is too small, then action will be replanned from closed loop instead. This ensures the adaptability and reliability in action selection under uncertainty, as training action selection.

Part 2: Description of an Experiment

During implementation, to find the optimal method, I compared open-loop, closed-loop, and hybrid learning methods over 200 epochs, as Figure 1 shows. I initially assumed that more epochs would benefit deeper networks (6 layers) for adequate learning. The trend in Figure 1 illustrates that the open-loop method showed the quickest reach, followed by the hybrid and then the closed-loop method. Re-planning at each step requires extra time, making the closed-loop method the slowest. The open-loop method, planning actions just once and extracting an action with O(1) complexity, should be the fastest, with the hybrid method in between. The hybrid method was chosen for its balance in navigating varied complexity of environments, as it encourages plan following for quicker navigation over "valleys" and prevents getting stuck on "mountains," as discussed in Part 1.

After confirming the hybrid method, I continued to experiment with different numbers of training epochs to fine-tune the model. It reveals that reducing the training to 100 epochs further optimized performance, as shown in Figure 2, achieving the best balance between speed and learning depth. Training for only 50 epochs might lead to underfitting due to inadequate training, resulting in slower goal achievement. Meanwhile, 200 epochs might suggest overfitting, reducing the model's generalizability and its ability to plan from slightly different states efficiently. In addition, I also developed CEM planning during implementation. However, CEM never reached the goal across all the tested environments, leading to its exclusion in the end.

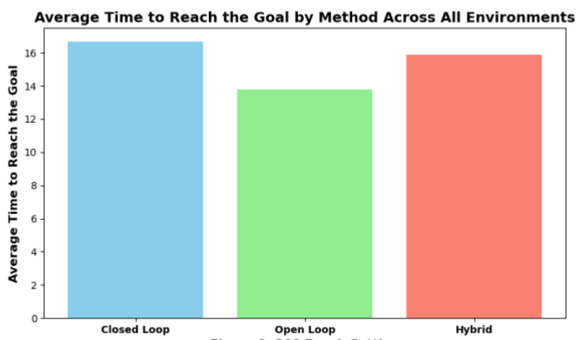

Figure 1: 200 Epoch Setting

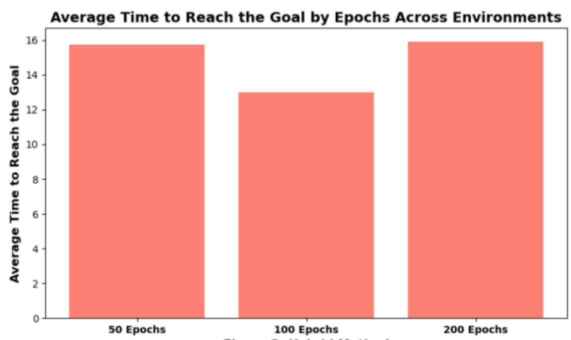

Figure 2: Hybrid Method